

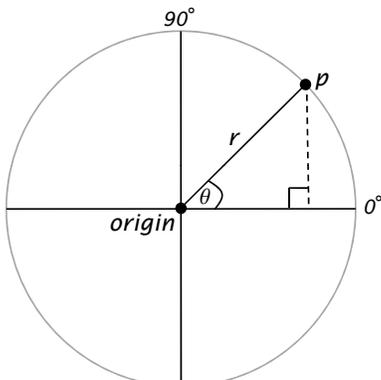
Review

- Array
 - `int[] diameters = new int[10];`
 - `diameters[0]`, `diameters[2]`, `diameters[9]`
 - `diameters.length`
- Indexing starts at 0
- A way to have a collection of variables instead of individual ones

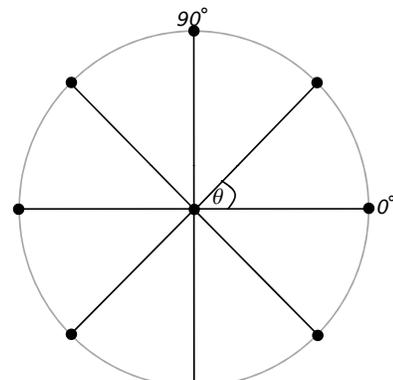
Built-in Array Functions

- `append(array, item)`
 - returns a new array expanded by one and add item to end
- `expand(array, newSize)`
 - returns a new array with size increased to newSize
- `shorten(array)`
 - returns a new array shortened by one
- `concat(array1, array2)`
 - returns a new array that is the concatenation of array1 and array2
- `subset(array, offset [, length])`
 - returns a subset of array starting at offset and proceeding for length (or end)
- `splice(array, value|array2, index) or`
 - returns a new array with value or array2 inserted at index
- `sort(array)`
 - returns a new array sorted numerically or alphabetically
- `reverse(array)`
 - returns a new array with all elements reversed in order

Trigonometry on a unit circle



Trigonometry on a unit circle



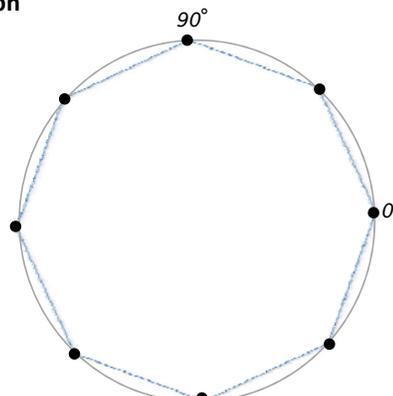
Drawing points along a circle

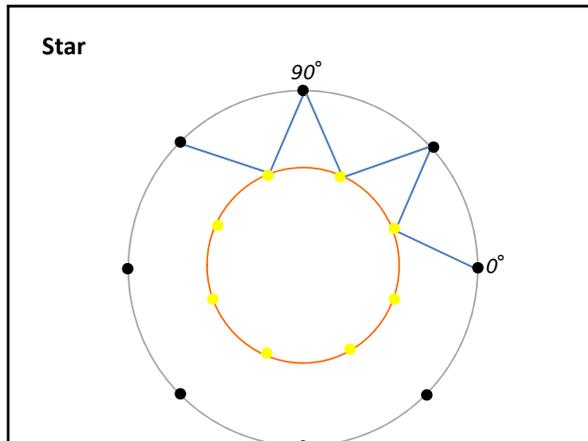
```
int steps = 8;
int radius = 20;
float angle = 2*PI/steps;

for (int i=0; i<steps; i++) {
    float x = sin(angle*i)*radius;
    float y = cos(angle*i)*radius;

    // draw a point every 1/8th of a circle
    ellipse(x, y, 10, 10);
}
```

Polygon



**Pop**

- A game that measures your balloon-popping skill.
- How it should work...
 - As game runs, randomly placed balloons inflate
 - When the player pops (clicks on) a balloon, 1 point is earned
 - Points are added up throughout the game duration
 - If one click is over top multiple balloons, all balloons pop and multiple points are earned
 - The game runs for 30 seconds, and then ends